Proyecto2-Pokemon

Doggos

09 de diciembre de 2019

Contents:

Nota: Bienvenido a la documentación del Proyecto 2 de la materia de Redes de Computadoras

CAPÍTULO 1

¿Cómo usar?

Nota: Primero inicializamos el servidor, y después los clientes pueden iniciar una conexión

- Para el servidor
  - Pasos previos para instalar la base de datos. Revisar archivo.
  - En una terminal, nos situamos en la ubicación del archivo *pokemonServer.py*
  - Para que se pueda establecer la comunicación satisfactoriamente, necesitamos ingresar la dirección IP al momento de iniciar el servidor. Por lo tanto, ejecutamos *./pokemonServer.py <IP>*
- Para el cliente
  - En una terminal, nos situamos en la ubicación del archivo *pokemonClient.py*
  - Este programa recibe como parámetros iniciales la dirección IP a través de la cual se quiere conectar, y el puerto. Por lo tanto, ejecutamos de la siguiente manera: *./pokemonClient.py <IP> <port>*

Nota: La dirección IP ingresada al ejecutar el cliente y el servidor debe ser la misma

CAPÍTULO 2

Programas involucrados

2.1 Cliente Pokemon Go!

Implementación de un cliente para el juego Pokemon Go! e interactúa directamente con el usuario

pokemonClient.**cerrarPorTimeout**(*soc*)
Cierre de sesión por tiempo de espera excedido

Parámetros **soc** (*Socket*) – Socket de la conexión

Devuelve Nada

pokemonClient.**cerrarSesion**(*soc*)
Cierre normal de sesión del usuario

Parámetros **soc** (*Socket*) – Socket de la conexión

Devuelve Nada

pokemonClient.**login**(*soc*)
Transfiere los datos al servidor para validar el acceso, y cierra el programa si los datos no son válidos

Parámetros **soc** (*Socket*) – Socket de la conexión

Devuelve Nada

pokemonClient.**main**()
Función principal

pokemonClient.**muestraPokemon**(*bytes*)
Despliega el pokemon asignado

Parámetros **bytes** – bytes de la imagen del pokemon a desplegar

Devuelve Nada

pokemonClient.**playPokemon**(*soc*)
Permite que el usuario juegue Pokemon Go

Parámetros **soc** (*Socket*) – Socket de la conexión

Devuelve Nada

2.2 Servidor Pokemon Go!

Implementación de un servidor para el juego Pokemon Go!

pokemonServer.**avisoTimeout**(*connection*)
Manda el mensaje de cierre de sesión al cliente por tiempo de espera excedido (timeout)

Parámetros **connection** (*Conexión*) – Conexión entre el cliente y el servidor

Devuelve Nada

pokemonServer.**cerrarSesion**(*connection*)
Cierre de sesión entre el servidor y el cliente al cual le pertenece la conexión

Parámetros **connection** (*Conexión*) – Conexión entre el cliente y el servidor

Devuelve Nada

pokemonServer.**clientThread**(*connection*, *ip*, *port*, *max_buffer_size=5120*)
Manejador del hilo que sostiene la conexión entre el servidor y un cliente

Parámetros

- **connection** (*Conexión*) – Conexión entre el servidor y el cliente que abrió el hilo
- **ip** (*Cadena*) – Dirección IP de la conexión
- **port** (*Entero*) – Puerto a través del cual el servidor mantiene la conexión con el cliente
- **max_buffer_size** (*Entero*) – Número máximo de bytes que puede recibir en un paquete del cliente

Devuelve Nada

pokemonServer.**giveAccess**(*connection*, *max_buffer_size=5120*)
Autentifica a usuarios registrados y proporciona acceso a la ejecución de la aplicación

Parámetros

- **connection** (*Conexión*) – Conexión entre el servidor y el cliente que abrió el hilo
- **max_buffer_size** (*Entero*) – Número máximo de bytes que puede recibir en un paquete del cliente

Devuelve int - Indicador de acceso permitido

pokemonServer.**main**()
Función principal.

pokemonServer.**playPokemonGo**(*connection*)
Método que simula el comportamiento del juego Pokemon Go

Parámetros **connection** (*Conexión*) – Conexión entre el servidor y el cliente

Devuelve Nada

pokemonServer.**start_server**(*ip_dir*)
Inicialización del servidor

Parámetros **ip_dir** (*Cadena*) – Dirección IP del socket al cual se va aconectar el servidor

Devuelve Nada

Índice de Módulos Python

p

Índice